

Get into Programming With JavaScript

Axle Barr

Functions

Function in the real world:

- Human body
- Vehicle
- Plants
- Computer
- HVAC

**THE
ACTIONABLE
PART OF
PROGRAMMING**

Daily Functions

My Daily Functions:

Wake up

Brush teeth

Eat bf

Go to work

Return home

Go biking

shower

**Think of all the
functions you
PERFORM in a
single day**

Daily Functions

My Daily Functions:

Wake up

Brush teeth

Eat bf

Go to work

Get sink fixed

Return home

Go biking

shower

**THINK OF ALL
THE FUNCTIONS
YOU PERFORM
IN A SINGLE DAY**

Daily Functions

My Daily Functions:

Wake up

Brush teeth

Eat bf

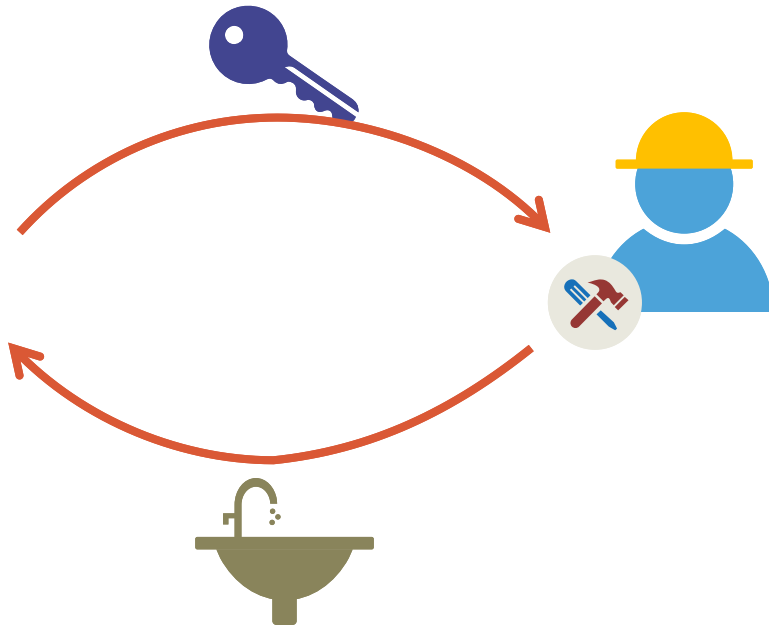
Go to work

~~getSinkFixed()~~

Return home

Go biking

shower



**THINK OF ALL
THE FUNCTIONS
YOU PERFORM
IN A SINGLE DAY**

FUNCTIONS AKA METHODS AKA SUB- ROUTINES

Facts about Functions

Blocks of code that perform a particular function

Input process output

The function will usually have a name and can be invoked anytime from any part of the program

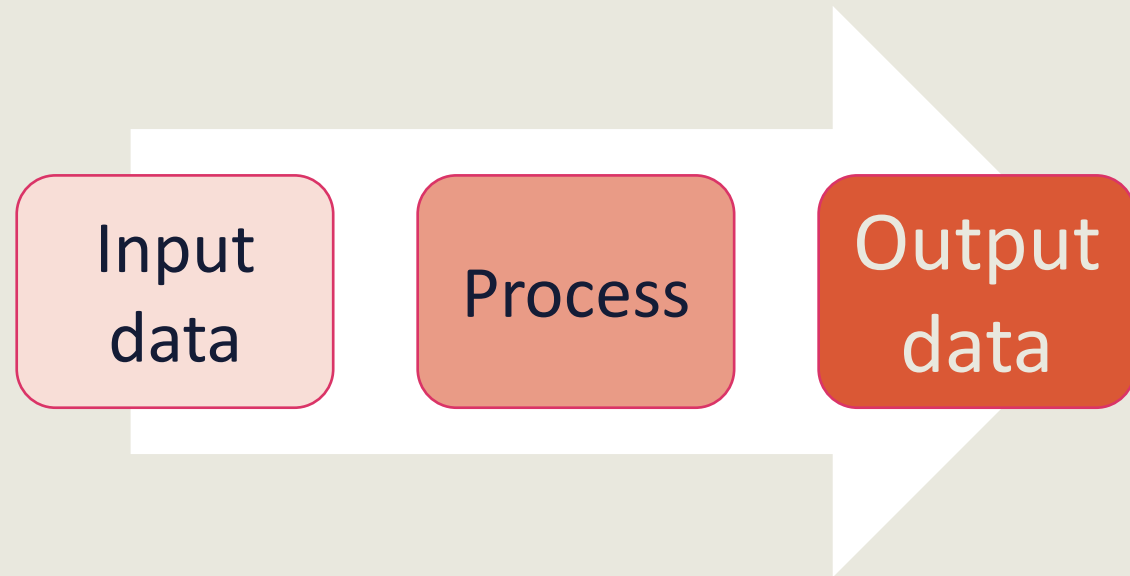
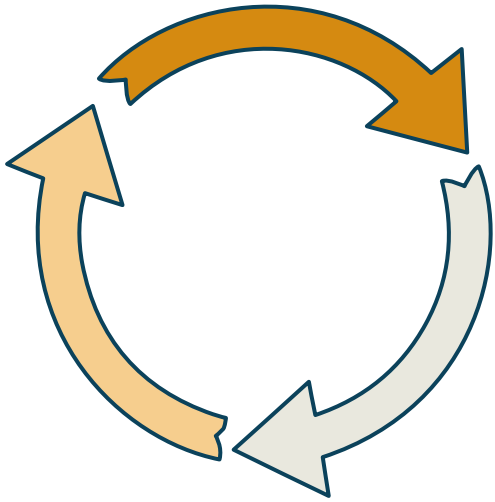
Any of the other programming structures can be part of a function

We can pass as many data points into the function, called **parameters**

Data or a result is passed back from the function, called the **return**

Functions are also called methods, sub-routines, modules

MINIATURE PROGRAMS



SIMPLE FUNCTION

```
function showOutput(){\n  document.getElementById("js_output").innerHTML=printOut;\n}
```


SIMPLE FUNCTION

function
keyword, start
of function

function name

```
function showOutput() {  
    document.getElementById("js_output").innerHTML=printOut;  
}
```

SIMPLE FUNCTION

```
function showOutput() {  
  document.getElementById("js_output").innerHTML=printOut;  
}
```

Bucket to pass
other variable
values to

Start of the
function body

SIMPLE FUNCTION

```
function showOutput(){\n  document.getElementById("js_output").innerHTML=printOut;\n}
```

Function body

End of the
function body

SIMPLE FUNCTION

```
function showOutput(){\n  document.getElementById("js_output").innerHTML=printOut;\n}
```

```
showOutput();
```



Function call

SIMPLE FUNCTION

```
function showOutput(x){  
    document.getElementById("js_output").innerHTML=printOut;  
}
```

Function
requires a
parameter

```
showOutput();
```

SIMPLE FUNCTION

```
function showOutput(x){  
  document.getElementById("js_output").innerHTML=printOut;  
}
```

Function
requires a
parameter

```
showOutput("Hello");
```

Pass the string
"hello" as a
parameter

SIMPLE FUNCTION

```
function showOutput(x){  
  let printOut = x;  
  document.getElementById("js_output").innerHTML=printOut;  
}
```

The value
inside of x is
assigned to
printOut

```
showOutput("Hello");
```

SIMPLE FUNCTION

```
let printOut="Hello";
```

Variable used inside the function is declared outside of the function

```
function showOutput(x){
```

```
    document.getElementById("js_output").innerHTML=printOut;  
}
```

```
showOutput();
```

We do not have to pass the value here as a parameter, notice empty parenthesis

Two functions

```
function doMath(x,y){  
  let z = x + y;  
  return z;  
}  
function showOutput(x,y){  
  printOut = doMath(5,6);  
  document.getElementById("js_output").innerHTML=printOut;  
}
```

Perform a job then send the results back to the line that called you

Call the doMath() function and pass to it what it needs

The result of the doMath() function will be assigned to printOut


SIMPLE FUNCTION

```
let printOut="";  
// do not change the top line  
function sayHello(x,y){  
    let z = x + " " + y;  
    return z;  
}  
// do not change the line below  
function showOutput(x,y){  
    printOut = sayHello("Hello","Skillsoft");  
    document.getElementById("js_output").innerHTML=printOut;  
}
```

Call the sayHello() function
and pass to it what it needs

MORE COMPLEX FUNCTIONS

```
let moreProducts = true, productCost=0.0, totalCart=0.0, moreItems=false;
while (moreProducts == true) {
    productCost = prompt("Enter price of product: ");
    productCost = parseFloat(productCost);
    totalCart = totalCart + productCost;
    moreItems = confirm("Do you have more items?");
    if(moreItems == false){
        moreProducts = false;
    }
}
```



The *else* part is missing!

MORE COMPLEX FUNCTIONS

```
let moreProducts = true, productCost=0.0, totalCart=0.0, moreItems=false;  
function getCartTotal ( ) {  
    while (moreProducts == true) {  
        productCost = prompt("Enter price of product: ");  
        productCost = parseFloat(productCost);  
        totalCart = totalCart + productCost;  
        moreItems = confirm("Do you have more items?");  
        if(moreItems == false){  
            moreProducts = false;  
        }  
    }  
}
```

MORE COMPLEX FUNCTIONS

```
let moreProducts = true, productCost=0.0, totalCart=0.0, moreItems=false;

function getCartTotal ( ) {

    while (moreProducts == true) {

        productCost = prompt("Enter price of product: ");
        productCost = parseFloat(productCost);
        totalCart = totalCart + productCost;
        moreItems = confirm("Do you have more items?");
        if(moreItems == false){
            moreProducts = false;
        }

    }

    return totalCart;
}
```

The purpose of this function is to give the calling function a total

```
function showOutput(){  
  printOut = getCartTotal();  
  document.getElementById("js_o  
utput").innerHTML=printOut;  
}
```

```
let moreProducts = true, productCost=0.0, totalCart=0.0, moreItems=false;  
function getCartTotal ( ) {  
  while (moreProducts == true) {  
    productCost = prompt("Enter price of product: ");  
    productCost = parseFloat(productCost);  
    totalCart = totalCart + productCost;  
    moreItems = confirm("Do you have more items?");  
    if(moreItems == false){  
      moreProducts = false;  
    }  
  }  
  return totalCart;  
}
```

Multiple Functions

```
function getCartTotal() {  
  while (moreProducts == true) {  
    productCost = prompt("Enter price of product: ");  
    productCost = parseFloat(productCost);  
    totalCart = totalCart + productCost;  
    moreItems = confirm("Do you have more items?");  
    if (moreItems == false) {  
      moreProducts = false;  
    }  
  }  
  return totalCart;  
}
```

Multiple Functions

```
function getCartTotal() {  
  while (moreProducts == true) {  
    productCost = prompt("Enter price of product: ");  
    productCost = parseFloat(productCost);  
    totalCart = totalCart + productCost;  
    moreItems = confirm("Do you have more items?");  
    if (moreItems == false) {  
      moreProducts = false;  
    }  
  }  
  return totalCart;  
}
```

```
function calculateTaxes() {  
  let afterTax = 0.0;  
  let beforeTax = getCartTotal();  
  afterTax = beforeTax * 1.08;  
  return afterTax;  
}
```


Multiple Functions

```
function getCartTotal() {  
  while (moreProducts == true) {  
    productCost = prompt("Enter price of product: ");  
    productCost = parseFloat(productCost);  
    totalCart = totalCart + productCost;  
    moreItems = confirm("Do you have more items?");  
    if (moreItems == false) {  
      moreProducts = false;  
    }  
  }  
  return totalCart;  
}
```

```
function calculateTaxes() {  
  let afterTax = 0.0;  
  let beforeTax = getCartTotal();  
  afterTax = beforeTax * 1.08;  
  return afterTax;  
}
```

```
function showOutput() {  
  printOut = calculateTaxes();  
  
  document.getElementById("js_output").innerHTML = printOut;  
}
```

Multiple Functions

```
function getCartTotal() {  
  while (moreProducts == true) {  
    productCost = prompt("Enter price of product: ");  
    productCost = parseFloat(productCost);  
    totalCart = totalCart + productCost;  
    moreItems = confirm("Do you have more items?");  
    if (moreItems == false) {  
      moreProducts = false;  
    }  
  }  
  return totalCart;  
}
```

```
function calculateTaxes() {  
  let afterTax = 0.0;  
  let beforeTax = getCartTotal();  
  afterTax = beforeTax * 1.08;  
  return afterTax;  
}
```

```
function showOutput() {  
  printOut = calculateTaxes();
```

```
document.getElementById("js_out  
put").innerHTML=printOut;
```

Multiple Functions

```
function getCartTotal() {  
  while (moreProducts == true) {  
    productCost = prompt("Enter price of product: ");  
    productCost = parseFloat(productCost);  
    totalCart = totalCart + productCost;  
    moreItems = confirm("Do you have more items?");  
    if (moreItems == false) {  
      moreProducts = false;  
    }  
  }  
  return totalCart;  
}
```

```
function calculateTaxes() {  
  let afterTax = 0.0;  
  let beforeTax = getCartTotal();  
  afterTax = beforeTax * 1.08;  
  return afterTax;  
}
```

```
function showOutput() {  
  printOut = calculateTaxes();
```

```
document.getElementById("js_out  
put").innerHTML=printOut;
```

Multiple Functions

```
function getCartTotal() {  
  while (moreProducts == true) {  
    productCost = prompt("Enter price of product: ");  
    productCost = parseFloat(productCost);  
    totalCart = totalCart + productCost;  
    moreItems = confirm("Do you have more items?");  
    if (moreItems == false) {  
      moreProducts = false;  
    }  
  }  
  return totalCart;  
}
```

```
function calculateShipping() {  
  let shipCharge = 0.0, cartTotal = 0.0;  
  cartTotal = getCartTotal();  
  if (cartTotal < 100) {  
    shipCharge = 5.0;  
  }  
  return shipCharge;  
}
```

```
function calculateTaxes() {  
  let afterTax = 0.0;  
  let beforeTax = getCartTotal();  
  afterTax = beforeTax * 1.08;  
  return afterTax;  
}
```

```
function showOutput() {  
  printOut = calculateTaxes() + calculateShipping();  
  document.getElementById("js_output").innerHTML = printOut;  
}
```

FLOWCHART TO SHOW FUNCTION/SUBROUTINE

